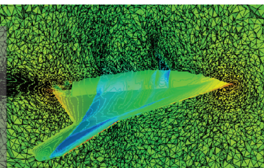
MESH ADAPTATION FOR SU2 WITH THE AMG LIBRARY

Victorien Menier*, Jean de BecdeLièvre*‡, Juan J. Alonso*

* Stanford University ‡ Inria



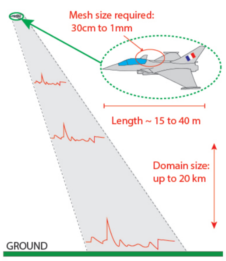
Stanford University Inria

This poster

presents the integration of AMG - the mesh adaptation software library by Inria - within the open-source SU2 suite developed at Stanford, and how we use this technology for on-going aircraft and nozzle optimization problems.

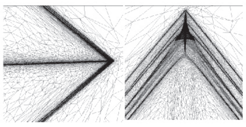
Why mesh adaptation?

The mesh is a crucial component of the CFD pipeline as it strongly impacts the accuracy of the solution and the total simulation time. Anisotropic mesh adaptation aims at optimizing the tradeoff between accuracy and computational time.



Example: Sonic boom prediction

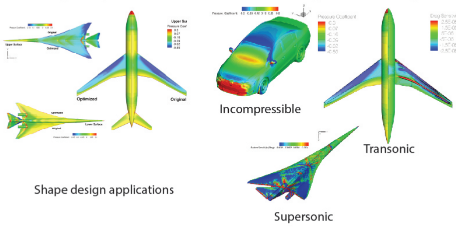
- Without mesh adaptation :
  1m precision : 200 Billion DoF
- Using mesh adaptation : 0.1 Billion DoF
- The problem complexity is decreased by 3 orders of magnitude just by modifying the discretization.



The SU2 suite

is an open-source collection of software for multi-physics simulation and design on unstructured meshes (i.e., CFD and beyond) developed in the Aerospace Design

- C++ / MPI based
- Python wrapping to perform complex analysis and design tasks
- Turbulence modeling: Spalart-Allmaras, Menter SST models
- Space integration: convective fluxes using central or upwind methods. Numerical schemes: JST, Roe, HLLC etc. Viscous fluxes using a finite volume method.
- Implicit/explicit time integration : implicit Euler scheme, local time stepping, global adaptive CFL value, etc.
- Acceleration techniques: nonlinear multigrid method (agglomeration), linelet preconditioning, etc.
- Sensitivity computation: continuous and discrete adjoint methodologies



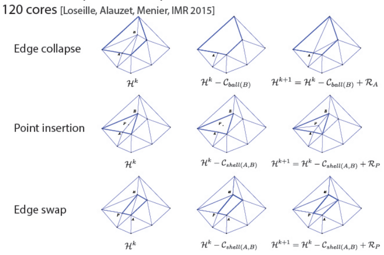
[SU2 Team, "SU2: An Open-Source Suite for Multiphysics Simulation and Design", AIAA Journal 2016]

AMG

is a mesh adaptation software library developed by the Gamma3 team at Inria. Its capabilities include anisotropic remeshing, metric computation, solution interpolation, and boundary layer mesh generation.

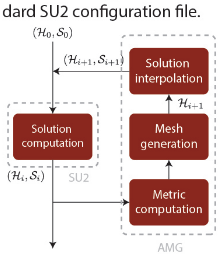
Capabilities:

- Volume and surface remeshing through local modifications [Loseille and Lohner, IMR 2009]
- Anisotropic remeshing using a single cavity-based operator [Loseille and Menier, IMR 2013]
- Metric computation from a solution field using a multi-scale metric [Loseille and Alauzet, SIAM 2010]
- Metric field correction using anisotropic gradation [Alauzet, FEAD 2010]
- Solution field interpolation from one mesh onto another [Alauzet and Mehrenberger, IJNME 2010]
- Parallel anisotropic mesh adaptation : 1Billion tetrahedra in less than 20 minutes on 120 cores [Loseille, Alauzet, Menier, IMR 2015]



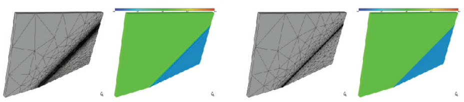
A python wrapper

is used to run the whole adaptive process. Flow solver and remeshing parameters are provided by the user through additional parameters in the standard SU2 configuration file.



- Mesh adaptation is an iterative process during which both the mesh and the solution are converged toward an optimal state.
- SU2 and AMG are called iteratively by a python wrapper, according to the parameters specified in the configuration file.
- Command line :

```
$ python mesh_adaptation.py -f config.cfg
```



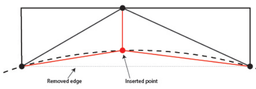
Iteration 1 Iteration 3

- Extract of the configuration file:

```
% ------- MESH ADAPTATION PARAMETERS ------

% Desired mesh sizes (appr. number of vertices)
ADAP_mesh_sizes= (1e6, 2e6)

% Number of sub-iterations for each mesh size
ADAP_SUBITE= (3, 3)

% Use an initial solution? (YES or NO)
ADAP_RESTART= NO
% Name of the initial mesh
ADAP_INI_MESH_FILE= m6_wing.meshb
% (if YES) name of the initial SU2 restart solution
ADAP_INI_RESTART_FILE= m6_wing_restart.solb
% (if YES) name of the initial sensor solution
ADAP_INI_SENSOR_FILE= m6_wing_mach.solb

% Surface reprojection (CAD/BACK_MESH/NONE)
ADAP_PROJ_METHOD= CAD
% If not NONE, specify the name of the cad model/back mesh file
ADAP_BACK_NAME= m6_wing_cad.iges

% Maximal edge sized allowed through the mesh adaptation
ADAP_HMAX= 70
% Minimal edge size
ADAP_HMIN= 0.00001
% Required mesh gradation
ADAP_HGRAD= 1.3
```

Point projection onto the CAD model

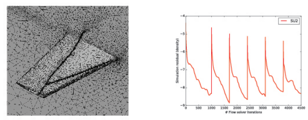
is crucial for the quality of the adapted final mesh. Two methods are available : GELite (Pointwise) and EGADS (MIT).



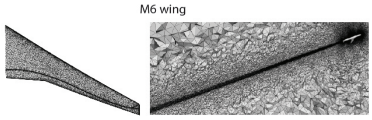
- GELite is developed by Pointwise. It is a 4th generation, non-manifold, surface and solid modeling, meshing, and computational geometry kernel written in C++.
- EGADS is an open-source code that provides a "bottom up" and/or Constructive Solid Geometry foundation for building Aircraft [Haimes and Drela, AIAA 2012].

The SU2/AMG interface was extensively tested

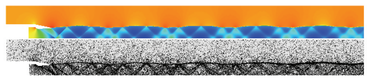
for various geometries and flow conditions.



M6 wing



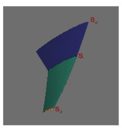
NASA CRM

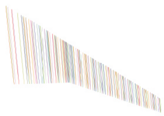


Nozzle flow (Mach number)

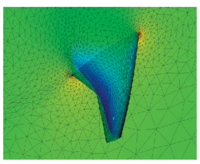
Optimal shape design of a wing:

a wing geometry is created from scratch using EGADS, then an optimization of the lift-to-drag ratio (L/D) based on the twist of the wing sections is performed using mesh adaptation in the loop.

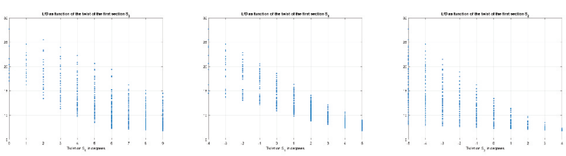




A CAD model is created from scratch using EGADS according to wing sections distributed along the span. The optimization parameters here are the twist value of sections S0, S1 and S2. The twist distribution over the wing is then obtained by linear interpolation.

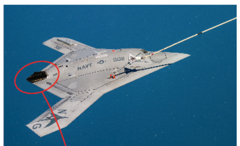


Mesh-adaptive flow computation. The simulation is stopped whenever the Drag and the Lift coefficients are converged by a given order of magnitude.

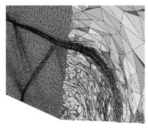


L/D as a function of the twist of each section.

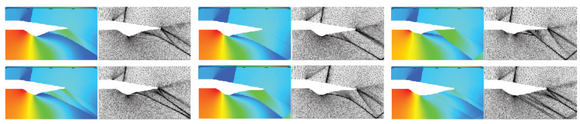
A complex Design Under Uncertainty (DUU) problem : Nozzle Aero-Thermal-Structural Design.



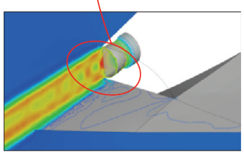
- Inspired by the X47-B
- Unmanned combat vehicle aircraft demonstrator
- Capable of carrier take-off and landing
- Complex nozzle shape integrated into the aft end of the vehicle
- Advanced materials and significant heat environment and thermal management issues
- Nozzle weight is a significant portion of the overall propulsion system weight

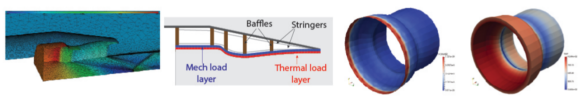


Adapted volume mesh



Hundreds or thousands simulations using as many nozzle shapes might be run during an optimization process: the robustness of the adaptive process is crucial.





Coupling with a FEM code [AEROS, FRG, Stanford]:
Mechanical (left) and thermal (right) stresses are computed using a two-layer wall design.